FRUIT FRESHNESS MEASUREMENT USING ML TECHNIQUES

B.Tech Project Report

By

Name of the B.Tech. Students
AKASH MUKHERJEE (GCECTB-R18-3035)
DEBMALYA SUR (GCECTB-R18-3007)
GOURAB CHATTERJEE (GCECTB-R18-3011)
SAYAN MONDAL (GCECTB-R18-3024)

Under Supervision of
Kingshuk Chatterjee

Department of Computer Sci. and Engineering

Government College of Engineering and Ceramic Technology
Kolkata

May 2022

FRUIT FRESHNESS MEASUREMENT USING ML TECHNIQUES

A Project Report

Submitted in partial fulfilment of the requirements for the award of the degree
of

Bachelor of Technology

In

Computer Sci. and Engineering

By

AKASH MUKHERJEE (GCECTB-R18-3035)
DEBMALYA SUR (GCECTB-R18-3007)
GOURAB CHATTERJEE (GCECTB-R18-3011)
SAYAN MONDAL (GCECTB-R18-3024)

Department of Computer Sci. and Engineering

Government College of Engineering and Ceramic Technology
Kolkata

May 2022

DECLARATION

We hereby declare that the project entitled **Fruit Freshness Measurement Using ML Techniques** was submitted for the B. Tech. (CSE) degree is our original work and the project has not formed the basis for the award of any other degree, diploma, fellowship, or any other similar titles.

Name and Roll No. of the Students	**Signature of the Students**
1.**AKASH MUKHERJEE (GCECTB-R18-3035)**	________________________
2.**DEBMALYA SUR (GCECTB-R18-3007)**	________________________
3.**GOURAB CHATTERJEE (GCECTB-R18-3011)**	________________________
4.**SAYAN MONDAL (GCECTB-R18-3024)**	________________________

Place: Kolkata

Date: 11/05/2022

	Government College of Engineering and Ceramic Technology **73, A. C. Banerjee Lane, Kolkata, West Bengal 700010**

••

BONAFIDE CERTIFICATE

Certified that this project report titled **FRUIT FRESHNESS MEASUREMENT USING ML TECHNIQUES** is the authentic work carried out by………..NAME OF THE STUDENT(Roll.Nos)...who carried out the project work under **my / our** supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

……………………………………………….. ……………………………………………

Kingshuk Chatterjee
SUPERVISOR
Assistant Professor
Department of Computer Science and Engineering
Government College of Engineering and
Ceramic Technology
Kolkata-700010

External Examiner

…………………………………………….
Dr.Kalpana Saha Roy
HEAD OF THE DEPARTMENT
Assistant Professor & Head
Department of Computer Science and Engineering
Government College of Engineering and
Ceramic Technology
Kolkata-700010

Abstract

This project presents a comprehensive analysis of a variety of fruit and vegetable images for freshness grading using deep learning. A number of models have been used in this project, including DenseNet121 and the Custom model. Fruit and vegetable decaying occur in a gradual manner, this characteristic is included for freshness grading by interpreting chronologically-related fruit decaying information.
The contribution of this project is to propose a novel neural network structure for fruit object locating and classification. It takes fruits and vegetables as an object and using its images we classify them into two categories bad and fresh.

Acknowledgment

We express our sincere gratitude towards our project mentor Kingshuk Chatterjee, Assistant Professor, Department of Computer Science and Engineering, Government College of Engineering and Ceramic Technology, Kolkata, West Bengal, for providing valuable guidance and constant encouragement throughout the project. We would also like to thank our HOD Mam Dr. Kalpana Saha Roy and Principal Sir Prof. Krishnendu Chakrabarty for providing us with this opportunity. It is our pleasure to record our sincere thanks to him for his constructive criticism and insight without which the project would not have shaped as it has.

We thank God for making all this possible, our parents and friends for their constant support and encouragement throughout the project work.

List of Figures

Figure	Page Number
Fig 2.1 Literature Survey	3
Fig 3.2.1 Examples of image from our dataset	5
Fig 3.2.2 Demonstration of our dataset	6
Fig 3.3.1.1 DenseNet-121 model architecture	8
Fig 3.3.2.1 Custom Model architecture	10
Fig 3.3.3.1 Error-vs-Complexity graph of Ridge Regressor	11
Fig 4.1.1 Training vs Validation loss and accuracy curves of DenseNet121	13
Figure 4.1.2 Confusion matrix of the validation set of DenseNet121	14
Figure 4.1.3 Confusion matrix of the test set of DenseNet121	14
Figure 4.1.4 Classification report of the validation set of DenseNet121	14
Figure 4.1.5 Classification report of the test set of DenseNet121	14
Figure 4.2.1 Training vs Validation loss and accuracy curves of FFCM	15
Figure 4.2.2 Confusion matrix of the validation set of FFCM	15
Figure 4.2.3 Confusion matrix of the test set of FFCM	15
Figure 4.2.4 Classification report of the validation set of FFCM	16
Figure 4.2.5 Classification report of the test set of FFCM	16
Figure 3.1 Mean Absolute Error Formula	17
Figure 4.3.2 Mean Squared Error Formula	17
Figure 4.3.3 Root Mean Squared Error Formula	18
Figure 4.3.4 Root Mean Squared Log Error Formula	18
Figure 4.3.5: R2 Score Formula	19
Figure 4.3.6: Adjusted R2 Formula	19
Figure 5.2.1 Data Flow Diagram	21
Figure 5.3.1 High Level Design	22
Figure 5.4.1 Home Page	22
Figure 5.4.2 Options for various Tasks	23
Figure 5.4.3 Classification Task	23
Figure 5.4.4 Regression Task	24
Figure 5.4.5 Combined Task	25

Table of Contents

	Title Page	i
	Declaration of the Students	iii
	Bonafide Certificate	iv
	Abstract	v
	Acknowledgment	vi
	List of Figures	vii
1.	**INTRODUCTION**	**1**
	1.1 Background 1.2 Fruit Freshness Grading 1.3 Project Motivation	1 1 2
2.	**LITERATURE SURVEY**	**3**
3.	**PROPOSED WORK**	**4**
	3.1 Objectives &Goals 3.2 Dataset 3.3 Model Building 3.3.1 DenseNet-121 Model Architecture 3.3.2 Custom Model Architecture 3.3.3 Ridge Regression Model	4 4 7 7 9 11
4.	**RESULTS / OUTPUTS**	**13**
	4.1 Output of DenseNet-121 4.2 Output of Custom Model 4.3 Output of Ridge Regression Model	13 15 16
5.	**SYSTEM ANALYSIS &DESIGN**	**20**
	5.1 System Specifications 5.2 DFD 5.3 Design and Test Steps 5.4 Web View	20 20 21 22
6.	**CONCLUSIONS AND SCOPE FOR FUTURE WORK**	**26**
	6.1 Conclusion 6.2 Future Work	26 26
7.	**REFERENCES**	**27**

1 Introduction

In this project, we will provide an overview of fruit freshness grading system, including the identification of the research problem.

1.1 Background

Fruits and vegetables are an important constituent of a person's daily diet and thus play a significant role in their lives. However, grading the freshness of the fruits and vegetables is a manual operation and is time-consuming. Also, not many people are skilled in the process of manual detection which leads to a large margin of error. Automated grading by using computerized approaches is believed to be the solution to this problem. In this current scenario people are getting more conscious about their health and trying to get fresh fruits and vegetables but are not able to correctly distinguish between fresh and bad.

Established research evidence shows that when deterioration of fruit and vegetable occurs, they go through a series of biochemical transformation that leads to changes in their physical conditions, e.g., visual features including colour and shape. Most of these features can be captured. It is expected that the computer vision-based approach is the most economical solution. Given the advancement of deep learning technology, grading algorithms should produce satisfactory accuracy.

1.2 Fruit Freshness Grading

For the categorization of the fruit and vegetables depending upon their freshness, we have used a Hedonic [9] scale rating. Each image is associated with a Hedonic [9] Rating given by the Panel of Food Technology experts. Hedonic [9] rating is a Quality Measurement technique measuring 3 qualities, namely 'Colour', 'Shape' and 'Texture'. Each quality can take a value from 1 to 9. The average of three values should be taken. If the average is greater than 4, then the image associated with the particular Hedonic [9] Rating is considered as "Fresh". Otherwise, the image will be considered as "BAD". Then the images are classified into two groups- "FRESH" & "BAD".

1.3 Project Motivation

We consider fruit freshness grading as one step of post-harvest assessments. Everyone tries to eat fresh fruits and vegetables as much as possible and the only way most of the common people can judge whether it is fresh or not is using manual inspection which is time consuming as well as it leaves a large error of margin and is also not scientifically proved. Moreover, many people might not even know the manual techniques and end up buying the stale ones. This motivated us to pursue this project as everyone wants fresh fruits and vegetables but is not able to distinguish it from the stale ones. As a result, it motivated us to find a way so that anyone can just use their phone’s camera to decide whether its fresh or not.

2 Literature Survey

We decided to do a literature survey to know the different models that we can use on our project.

The following table shows the various works being done in the area of fruit, vegetable classification or grading, the models being used and the accuracy achieved.

Author Name	**Fruit / Vegetable Used**	**Method used and Accuracy achieved**
Jahanbakhshi et al .[1]	Classified carrots based on their size to control waste	Artificial Neural Networks - 98.5% Support Vector Machine - 89.62%
Izadi et al .[2]	Graded tomatoes based on their size,shape and texture	Neural fuzzy networks (ANFIS) - 81%
Kheiralipour et al .[3]	Categorized cucumbers based on their size,shape and texture	Artificial Neural Networks - 97.1% Linear Discriminant Analysis - 91.2% Quadratic Discriminant Analysis - 93.1%
Wang et al. [4]	Classified fruits	Convolutional Neural Networks - 95.67%
Jahanbakshi et al.[5]	Classified lemons based on their imperfections	Convolutional Neural Networks - 100%
Chowdhury et al.[6]	Recognized 10 different types of vegetables based on colour and texture	Convolutional Neural Networks - 96.55%
Danti et al.[7]	Classified 10 different types of leafy vegetables	BP Neural Networks - 96.40%
Yuan et al.[8]	Created a database named Food-SK consisting of 2500 food images selected from three popular image sets for food recognition:	Fine Tuned GoogLeNet -99.2%

Figure 2.1 Literature Survey

3 The Proposed Work

3.1 Objectives & Goals

- The customers mainly use manual inspection techniques to distinguish between fresh and stale fruits and vegetables.
- This process is very time-consuming and we cannot determine the actual quality and there may be even some naive buyers who can't distinguish and end up buying the stale ones.
- In this project, we aim to automate the whole process so anyone can just go to the market and distinguish the fresh ones from the stale ones using their smartphone cameras.
- We aim to measure the freshness of the fruit utilizing a Hedonic [9] scale rating.
- Using the measurement, we can categorize the fruits and vegetables into two categories namely fresh and bad.
- Utilizing the categorization, we can determine whether it is consumable or not.

3.2 Dataset

In our project, we didn't use any readily available or existing dataset, rather we have created the dataset on our own. We selected Amla for our dataset because it is easily available at the market, the price is also lower compared to other fruit items, and being a rich source of vitamin-c, it is very useful especially in the COVID situation.

- We have image files and one CSV file containing Hedonic [9] ratings as our dataset.
- Seven Amlas of fine quality were collected.
- Those Amlas were observed for 15 days. Those were kept in the same environmental condition.
- The picture of each Amla was captured thrice a day.
- The time of capturing photos were 10 AM, 4 PM, and 10 PM.
- 21 images were captured per day for our dataset.
- There are 283 images of amla in total.

Figure 3.2.1 Examples of images from our Dataset

The process was started from 4th May 2021 onwards. We took 3 pictures of each amla per day. So, in total, we captured 21 images per day. 315 images should be there, but there are 283 images in our dataset. The rest of the images got eliminated due to the worst quality. After capturing the images and eliminating inappropriate images, the rest images are then renamed in a particular format, in order to easy accessibility with coding.

The other part of our dataset is the Hedonic [9] Rating. Each image is associated with a Hedonic [9] Rating given by the Panel of Food Technology experts. **Hedonic [9] rating is a Quality Measures technique**. 3 quality measures, namely 'Colour', 'Shape' and 'Texture' are there. Each quantity can take a value from 1 to 9, where "9" indicates the best condition and "1" stands for the worst condition. Each parameter should have an individual value. The average of three values should be taken. **If the average is greater than 4, then the image associated with the particular Hedonic [9] Rating is considered as "Fresh". Otherwise, the image will be considered as "BAD"**. Then the images are classified into two groups- "FRESH" & "BAD".

Image	Time &Date	Shape	Colour	Texture	Average	Status
	08/05/2021 4 pm Sample-1	8.5	8	8	8.2	Fresh
	18/05/2021 10 pm Sample-1	2	2	2	2	Bad

Figure 3.2.2 Demonstration of the dataset

So, in the above table, the first image was captured on the 8th day of May at 4 pm. This is sample-1. The shape rating of the image is 8.5. Texture and Colour ratings are 8.5 & 8 respectively. So, the average of these three parameters is 8.2, which is greater than 4. Hence, this particular image is classified as "Fresh". On other hand, the second image was captured on 18th May at 10 pm. It is sample-1. The value of Shape, Texture & Colour ratings are 2, 2 & 2 respectively. Since, the average of these three fields is 2, which is lesser than 4. That implies the corresponding item will be considered as “Bad” and it is inconsumable.
So, in our dataset, we have image files and one CSV file containing Hedonic [9] Rating. Each image is associated with one Hedonic [9] rating, better to say one row of the CSV file.

3.3 Model Building

After creating our dataset, we did some data pre-processing to make the data ready before insertion into Deep Learning Models. We aim to build a CNN model to do classification, we want to build models using Transfer Learning and custom models of Convolutional Neural Networks. We are going to apply our model created to classify the fruit items as FRESH or BAD.

3.3.1 DenseNet-121 Model Architecture

Input Layer: Here we will give an input of (150, 150, 3) sized image to the model and the output of this layer will be (150, 150, 3)
Functional Layer: It is a combination of multiple layers, first we will do a max pooling on the current layer and then we will create two threads, on one thread we will do the following procedures sequentially:
- Batch normalization
- Applying an activation function (ReLU is preferred)
- Then applying a convolution 2D layer

we will follow these sequences multiple times, and lastly, the output of this thread will be concatenated with another thread. In this model we got (4, 4, 1024) as an output from this layer.
Flatten Layer: The main purpose of this layer is to flat the nodes of the previous layer, so the total number of nodes in this layer will be the simple multiplication of the output of its previous layer. Here, the input to this layer is (4, 4, 1024) thus the output from this layer will be = 4 x 4 x 1024 = 16384
Dense Layer: The main purpose of this layer is to make a dense connection between the input nodes and the output nodes. A dense connection means each node of its input layer will be connected with each node of the output layer, i.e. this layer will create a complete bipartite graph-like structure. In this model, we have got 128 nodes as an output from this layer.
Dropout Layer: Dropout is a technique used to prevent a model from overfitting. Dropout works by randomly setting the outgoing edges of hidden units (neurons that make up hidden

layers) to 0 at each update of the training phase. In this model, we have got 128 nodes as an output from this layer.

Dense Layer: It will be the last layer of our model. As an input to this layer, we have 128 nodes and this layer will simply make a complete connection with the 2 output nodes in this layer and those two output nodes will be the output of the whole model, more accurately one node will be standing for "FRESH" and another node will be considered as "BAD" conditioned fruits.

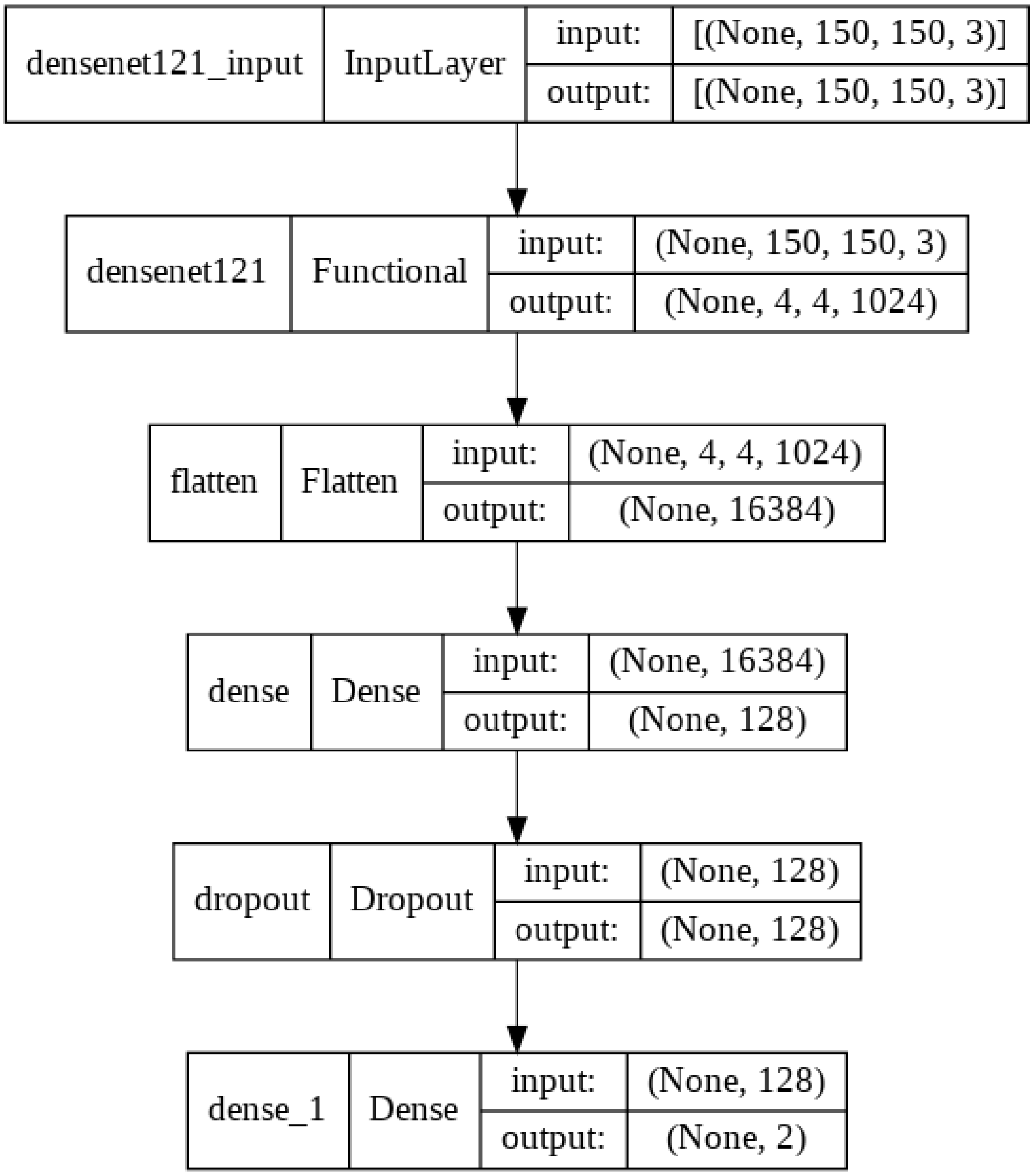

Figure 3.3.1 DenseNet-121 Model architecture

3.3.2 Custom Model Architecture

Input layer, the input shape is (224, 224, 1) and the output shape is also (224, 224, 1), which is then passed through Convolution2D layer with a filter of (3,3) and the input shape is (224, 224, 1) and output shape is (222, 222, 64) and in it Relu is used as an activation function. It is then passed through MaxPooling2D layer with a pool size of (3, 3) and the input shape is (222, 222, 64) and output shape is (222, 222, 64). It is then passed through Convolution2D layer with a filter of (3,3) and the input shape is (74, 74, 64) and output shape is (72, 72, 32) and in it Relu is used as an activation function. It is then passed through MaxPooling2D layer with a pool size of (2, 2) and the input shape is (72, 72, 32) and output shape is (36, 36, 32). It is then passed through Convolution2D layer with a filter of (3,3) and the input shape is (36, 36, 32) and output shape is (34, 34, 64) and in it Relu is used as an activation function. It is then passed through MaxPooling2D layer with a pool size of (2, 2) and the input shape is (34, 34, 64) and the output shape is (17, 17, 64). The data is then flattened where the input shape is (17, 17, 64) and the output shape is (18496), which is then passed through a dense layer where the input shape is (18496) and output shape is (128) and here Relu is used as an activation function and dropout (p=0.5) is used. It is then passed through a dense layer where the input shape is (128) and output shape is (64) and here Relu is used as an activation function and dropout (p=0.3) is used. It is then passed through a dense layer where the input shape is (64) and output shape is (1) and here Sigmoid is used as an activation function

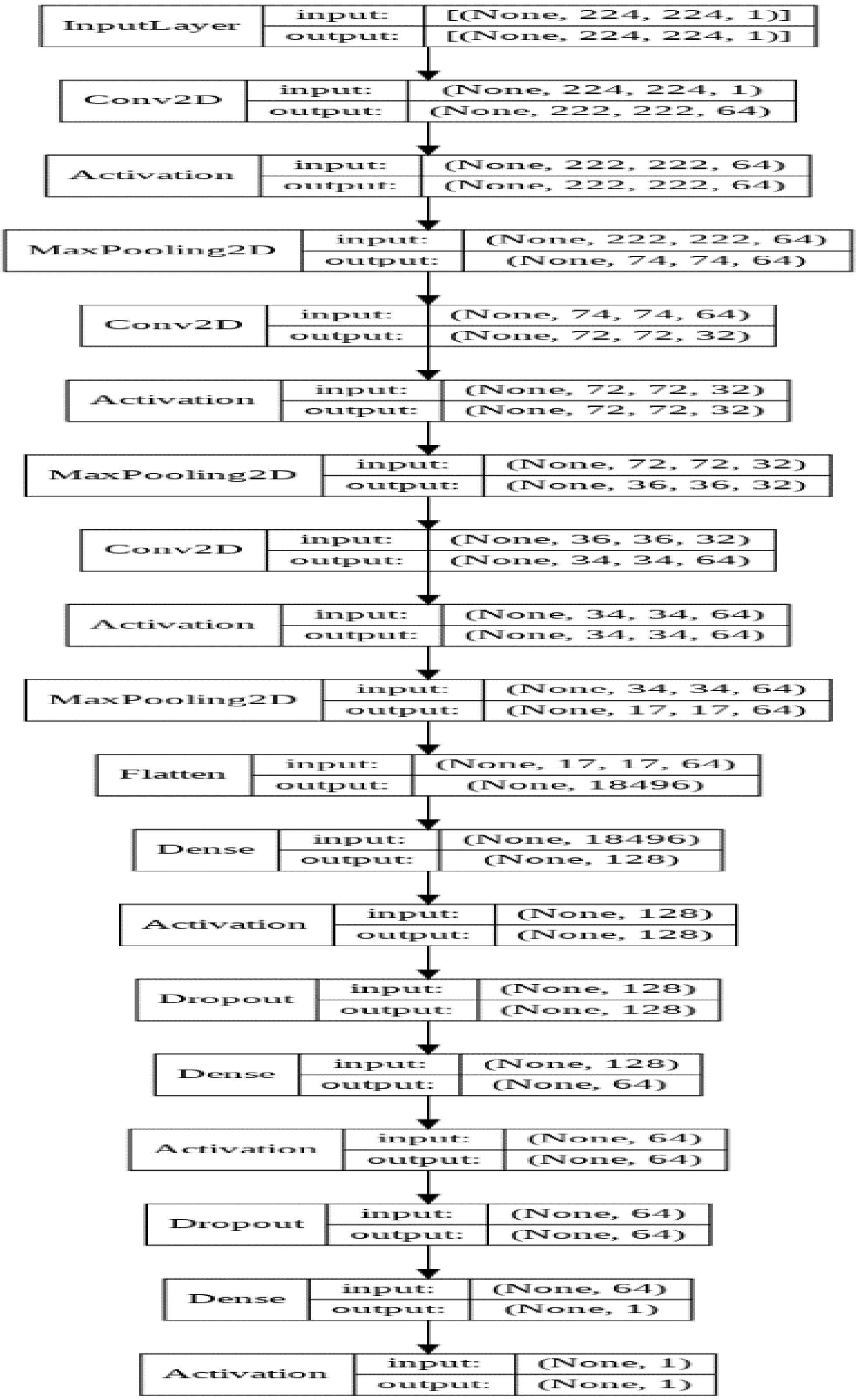

Figure 3.3.2.1 Custom Model architecture

3.3.3 Ridge Regressor

Ridge Regression is a special type of model tuning procedure. It is essentially used to analyse data that are suffering from multicollinearity. Multicollinearity means correlations between predictor variables. The mentioned method performs
L2 regularization techniques. It adds the Squared Magnitude of the coefficient as the penalty term to the loss function. Ridge Regression is a special way that creates a parsimonious model when the number of predictor variables in a set exceeds the number of observations. Alternatively, it is also very useful when data has multicollinearity. Ridge Regressor doesn't require unbiased estimators.
When there are some issues because of multicollinearity, least-squares are unbiased, and variances are large, which results in the predicted values being far away from the actual values.

The cost function of Ridge Regression is given here:

Cost Function = Minimum(**||y-x(θ)||2+ λ||θ||**2)

Here, is the Penalty term. It is denoted by an Alpha parameter in the Ridge Function. We can control the penalty term by changing the values for alpha. That means, bigger the values of alpha, the higher is the penalty and therefore the magnitude of coefficients will be reduced.

Since Ridge Regressor shrinks the parameters, this is used to prevent multicollinearity. It reduces the model complexity by shrinking the coefficients.

The Error vs Complexity scenario of the Ridge Regression is given below:

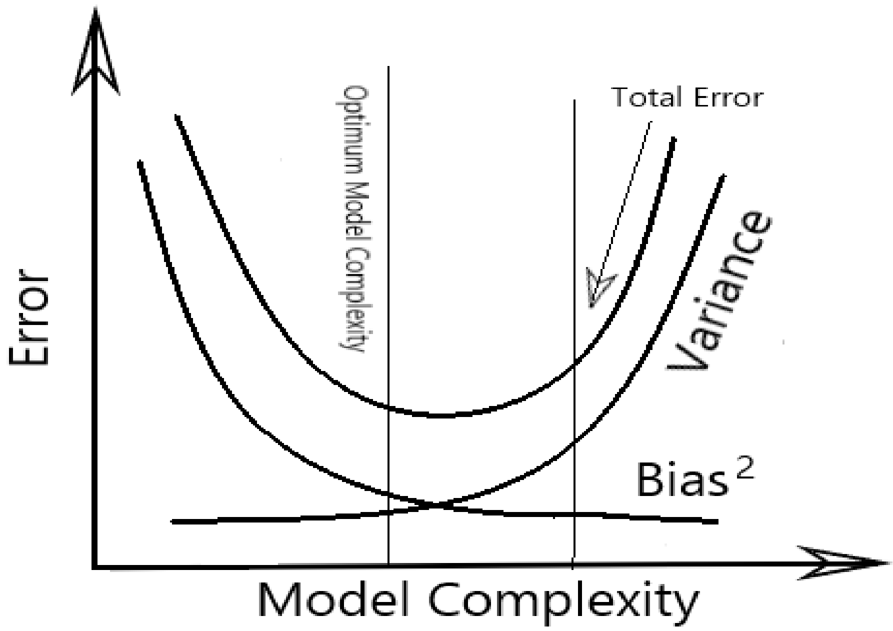

Figure3.3.3.1: - Error-vs-Complexity graph of Ridge Regressor

This is the Error Vs Model Complexity graph for Ridge Regressor. We can see the Variance and the Square of Bias in the above graph. We can determine Total Error and the Optimum Model Complexity through this graph.

4. Results and Outputs

Here we tried out two different models, one using transfer learning and the name of the model is **Dense Net 121**, and another was our custom model named as Fruit Freshness Classification Model in short **FFCM.** Here are the results of those two models.

4.1 Output of DenseNet121

The model's training and validation loss and accuracy curves are given below

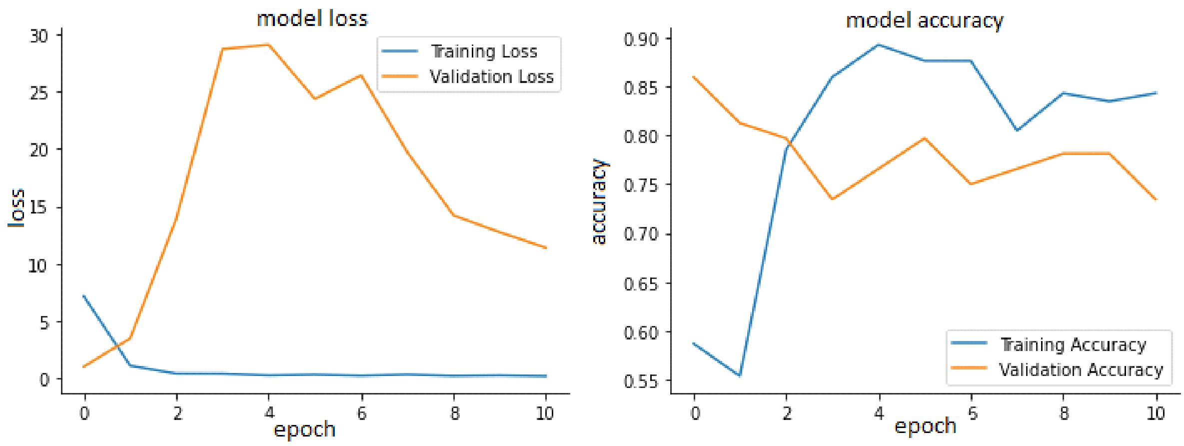

Figure 4.1.1 Training vs Validation loss and accuracy curves of DenseNet121

The confusion matrices of validation and test sets are given below

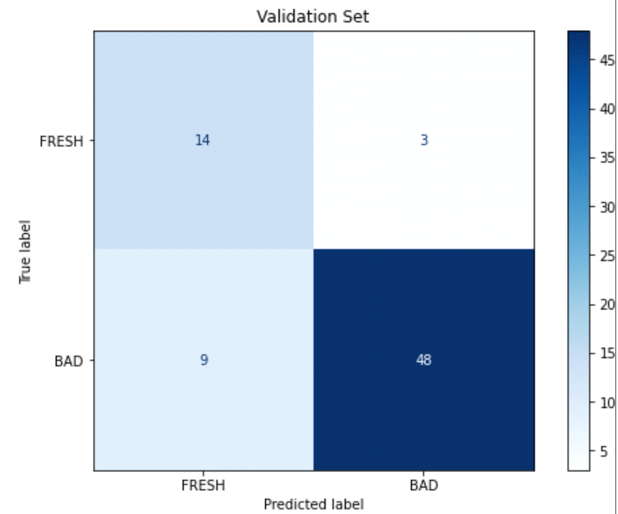

Figure 4.1.2 Confusion matrix of the validation set of DenseNet121

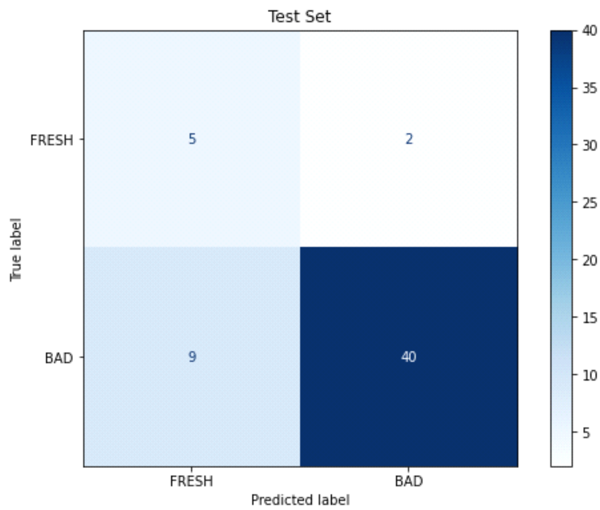

Figure 4.1.3 Confusion matrix of the test set of DenseNet121

The classification reports of the validation and test set are given below

	precision	recall	f1-score	support
BAD	0.61	0.82	0.70	17
FRESH	0.94	0.84	0.89	57
accuracy			0.84	74
macro avg	0.77	0.83	0.79	74
weighted avg	0.86	0.84	0.85	74

Figure 4.1.4 Classification report of the validation set of DenseNet121

	precision	recall	f1-score	support
BAD	0.36	0.71	0.48	7
FRESH	0.95	0.82	0.88	49
accuracy			0.80	56
macro avg	0.65	0.77	0.68	56
weighted avg	0.88	0.80	0.83	56

Figure 4.1.5 Classification report of the test set of DenseNet121

4.2 Output of FFCM

The model's training and validation loss and accuracy curves are given below

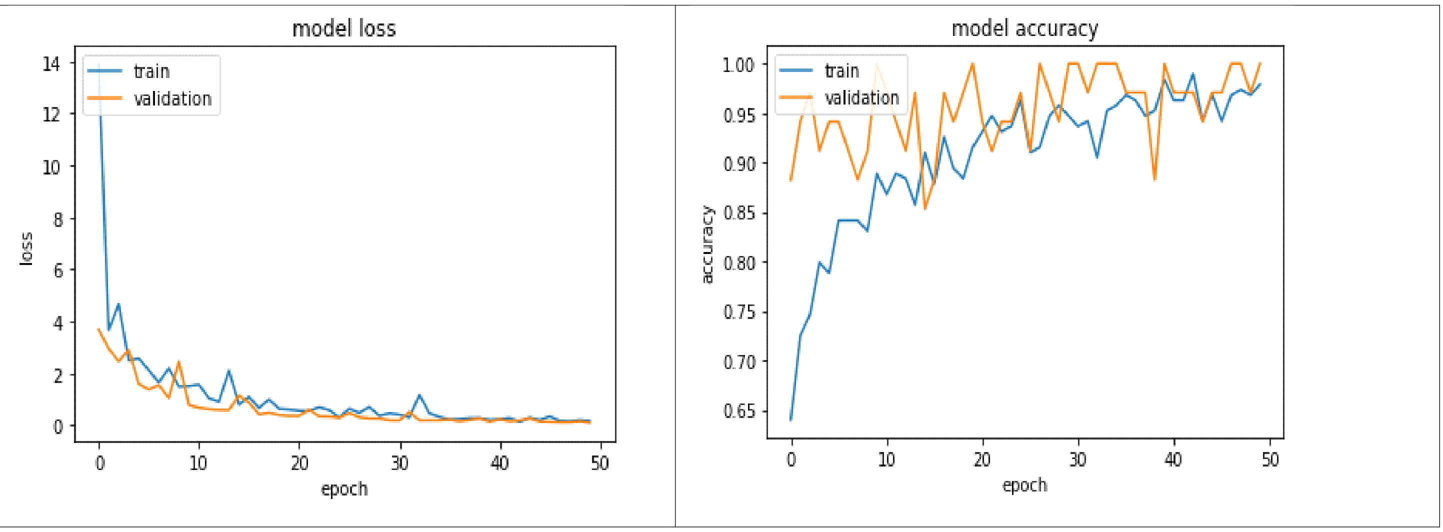

Figure 4.2.1 Training vs Validation loss and accuracy curves of FFCM

The confusion matrixes of validation and test sets are given below

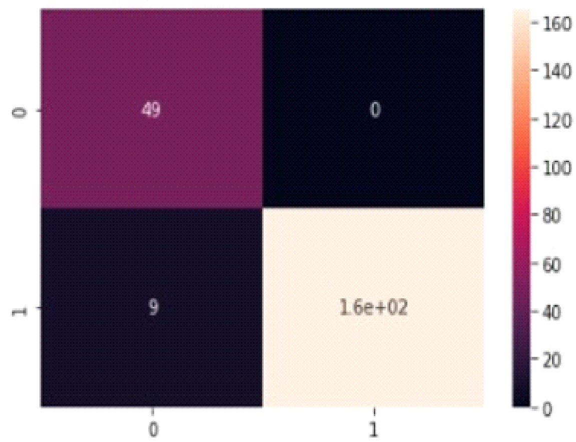

Figure 4.2.2 Confusion matrix of the validation set of FFCM

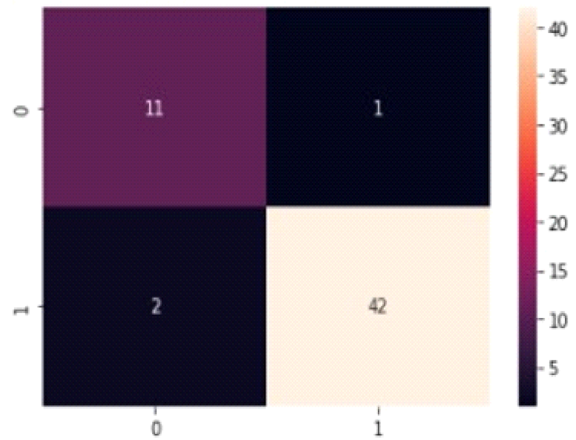

Figure 4.2.3 Confusion matrix of the test set of FFCM

The classification reports of the validation and test set are given below

	precision	recall	f1-score	support
BAD	0.84	1.00	0.92	49
FRESH	1.00	0.95	0.97	174
accuracy			0.96	223
macro avg	0.92	0.97	0.94	223
weighted avg	0.97	0.96	0.96	223

Figure 4.2.4 Classification report of the validation set of FFCM

	precision	recall	f1-score	support
BAD	0.85	0.92	0.88	12
FRESH	0.98	0.95	0.97	44
accuracy			0.95	56
macro avg	0.91	0.94	0.92	56
weighted avg	0.95	0.95	0.95	56

Figure 4.2.5 Classification report of the test set of FFCM

4.3 Output of FFRM

- **Mean Absolute Error: -** Absolute Error is nothing but the total error in our measurement. The Mean Absolute Error is measured by summing up all the absolute errors and dividing them by the total number of errors.

Average MAE = (Sum Of($|Y_i - X_i|$) / (Number of Errors) + Sum Of($|Y_j - X_j|$) / (Number of Errors) + Sum Of($|Y_k - X_k|$) / (Number of Errors)) /3

Here, i denotes “Shape”, j denotes “Color” and k denotes the “Texture”.

Figure.3.1: Mean Absolute Error Formula

The Average Mean Absolute Error for our model = 0.006478328295644295

- **Mean Squared Error: -** It is the mean of squared difference between target values and model predictions. We can measure the quality of an estimator through the term Mean Squared Error.

$$\text{Average MSE} = \left(\frac{1}{n} \sum_{i=1}^{n} (Y_i - \widehat{Y}_i) + \frac{1}{n} \sum_{i=1}^{n} (Y_j - \widehat{Y}_j) + \frac{1}{n} \sum_{i=1}^{n} (Y_k - \widehat{Y}_k) \right) / 3$$

Here, n = total no of samples, Y represents the target values and $\widehat{Y}$ represents the model predictions and i denotes "Shape", j denotes "Color" and k denotes the "Texture".

Figure 4.3.2: Mean Squared Error Formula

The Average Mean Squared Error for our Model = 0.0016195524146293838

- **Root Mean Squared Error: -** It is the squared root of the mean of squared differences between the target values and model predictions. This term shows the differences between the observed or actual values and the predicted values.

$$\text{Average RMSE} = \left(\sqrt{\frac{\sum_{i=1}^{n} ((Actual)_i - (Predicted)_i)^2}{n}} + \sqrt{\frac{\sum_{j=1}^{n} ((Actual)_j - (Predicted)_j)^2}{n}} + \sqrt{\frac{\sum_{k=1}^{n} ((Actual)_k - (Predicted)_k)^2}{n}} \right) / 3$$

Here, i denotes "Shape", j denotes "Color" and k denotes the "Texture".

Figure 4.3.3: Root Mean Squared Error Formula

The Average Root Mean Squared Error of our model = 0.0402436630369227

- **Root Mean Squared Log Error: -** It is the squared root of the mean of squared differences between the log transferred target values and the log transferred model predictions.

$$\text{Average RMSLE} = \left(\sqrt{\frac{\sum_{i=1}^{n} (log(Actual)_i - log(Predicted)_i)^2}{n}} + \sqrt{\frac{\sum_{j=1}^{n} (log(Actual)_j - log(Predicted)_j)^2}{n}} + \sqrt{\frac{\sum_{k=1}^{n} (log(Actual)_k - log(Predicted)_k)^2}{n}} \right) / 3$$

Here, i denotes “Shape”, j denotes “Color” and k denotes the “Texture”.

Figure 4.3.4: Root Mean Squared Log Error Formula

The Average Root Mean Squared Log Error of our model = 3.2128027275890076

- **R2 Score: -** It indicates the degree of interrelation. It says about the variation of the dependent variables from independent variables.

Average R Squared = (A/B + C/D + E/F) / 3

A= $\sum_{i=1}^{n} (\widehat{Y_i} - \overline{Y})^2$. It is the sum of Squared Regression of a model.

B= $\sum_{i=1}^{n} (Y_i - \overline{Y})^2$. It is the total variance in the data.

$\overline{Y}$ is the mean of Y values and $\widehat{Y_i}$ is the predicted value of observation i for Y.

C= $\sum_{j=1}^{n} (\widehat{Y_j} - \overline{Y})^2$. It is the sum of Squared Regression of a model.

D= $\sum_{j=1}^{n} (Y_j - \overline{Y})^2$. It is the total variance in the data.

E= $\sum_{k=1}^{n} (\widehat{Y_k} - \overline{Y})^2$. It is the sum of Squared Regression of a model.

F= $\sum_{k=1}^{n} (Y_k - \overline{Y})^2$. It is the total variance in the data.

Here, i denotes “Shape”, j denotes “Color” and k denotes the “Texture”.

Figure4.3.5 R2 Score Formula

The Average R2 score of our model = 0.9995107898512906

- **Adjusted R2: -** It is the modified version of the R Squared score. It is adjusted against the number of predictors.

$$\text{Adjusted R2} = 1 - \frac{(1-R2)(N-1)}{N-P-1}$$

Here, N = Total sample Size, P = Number of Independent Variables.

Figure 4.3.6 Adjusted R2 Formula

The Adjusted R2 score of our model = 0.9994843460594685

5 System Analysis & Design

5.1 System Specifications

Parameters	Google Colab
GPU	Nvidia K80 / T4
GPU Memory	12GB / 16GB
GPU Memory Clock	0.82GHz / 1.59GHz
Performance	4.1 TFLOPS / 8.1 TFLOPS
No. CPU Cores	2
Available RAM	12GB (upgradable to 26.75GB)
Disk Space	358GB

5.2 Data Flow Diagram

We have used the Iterative Enhancement mechanism to build our system. In this concept, one software can be divided into several parts or modules and each module can be developed incrementally. After each iteration, an updated version can be generated. A complete concept or flow diagram is given in the Figure.

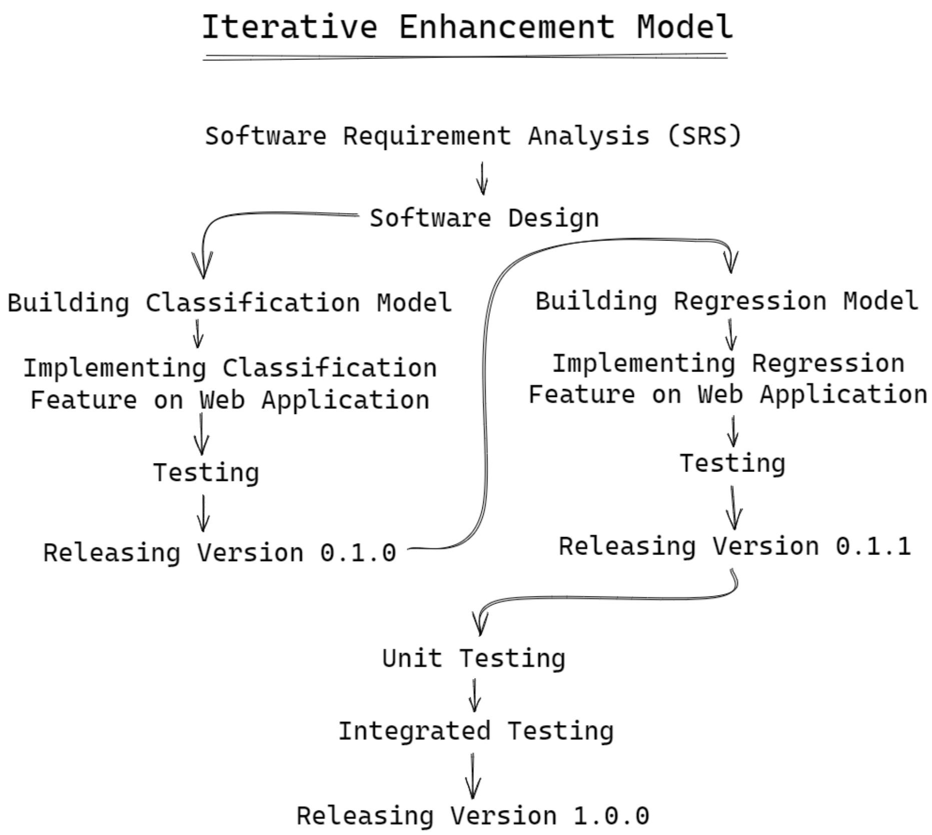

Figure 5.2.1 Data Flow Diagram

5.3 Design & Test Steps: -

Now, let's look into the internal mechanism of our system. We have stored the Regression and Classification model in the Database System. When the user will upload an image, it will be saved temporarily in the cloud. After running the requested model, the result will be generated and the user will get the required result. A diagram showing the High-Level Design is given in Figure.

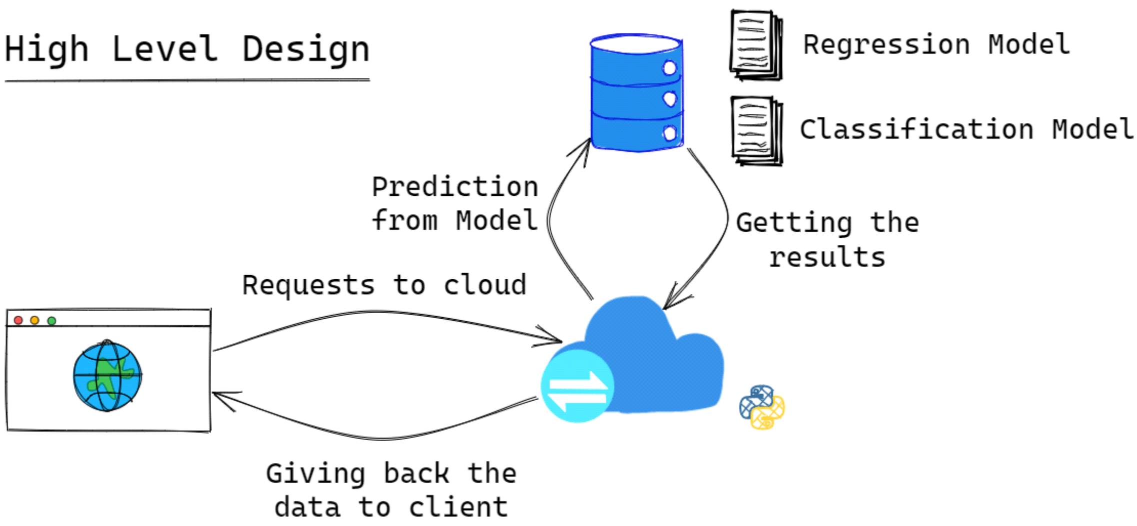

Figure 5.4.1 High Level Design

5.4 Web View

We have used Flask as the Backend and Bootstrap as the Frontend. There are 3 parts to our Web App. The first one is Classification Task. after uploading an image, our system will say whether the corresponding fruit item is consumable or not. The second one is the Regression part. This part will predict the 3 different ratings of the Hedonic System. And the average of these 3 fields Colour, Shape and Texture ratings will also be there. The final part is the Combined task of Classification and Regression. Some snapshots of the web app are given below:

- **Home Page: -** The Home Page of our Web App looks like the one below.

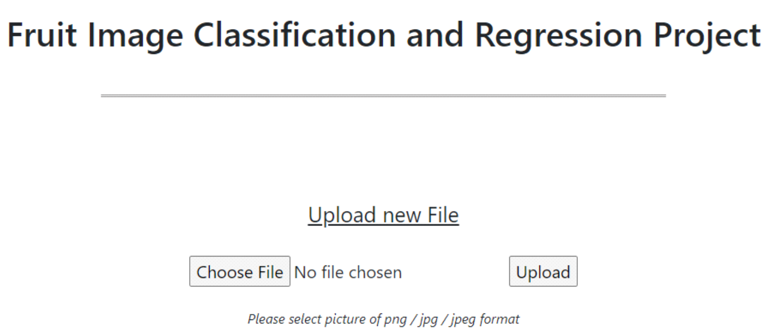

Figure 5.5.1 Home Page

- **After Successfully uploading an image by the user: -** When the user uploads an image successfully, then it will be sent to the server. And our App will look like the one below. We will have three options- Previewing the uploaded image, Classification, Regression and finally Combination of Classification and Regression tasks. Figure represents the above discussions.

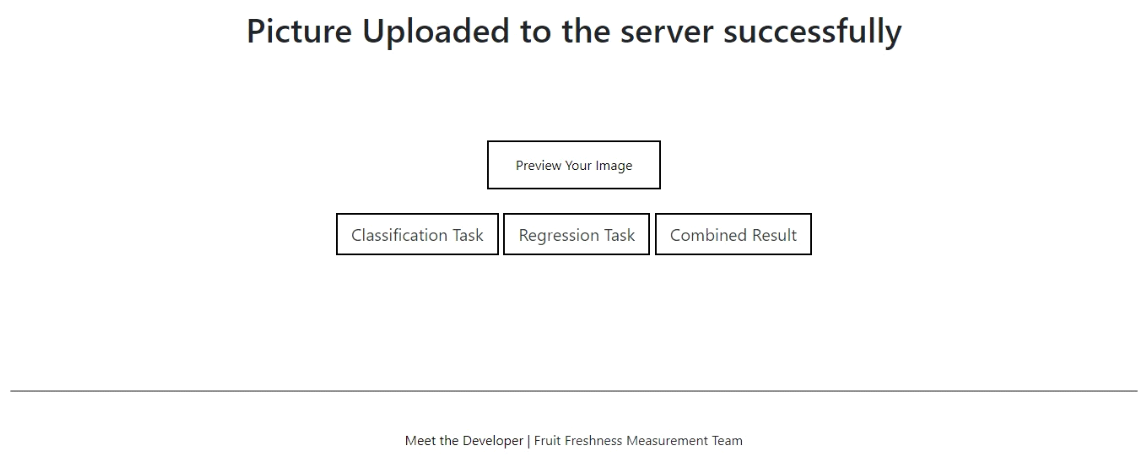

Figure 5.5.2 Options for various Tasks

- **Classification Report: -** When the user wants to perform Classification, then our system will say like figure.

Figure 5.5.3 Classification Task

- **Regression Report: -** After performing the Regression task, the page will look like the one below.

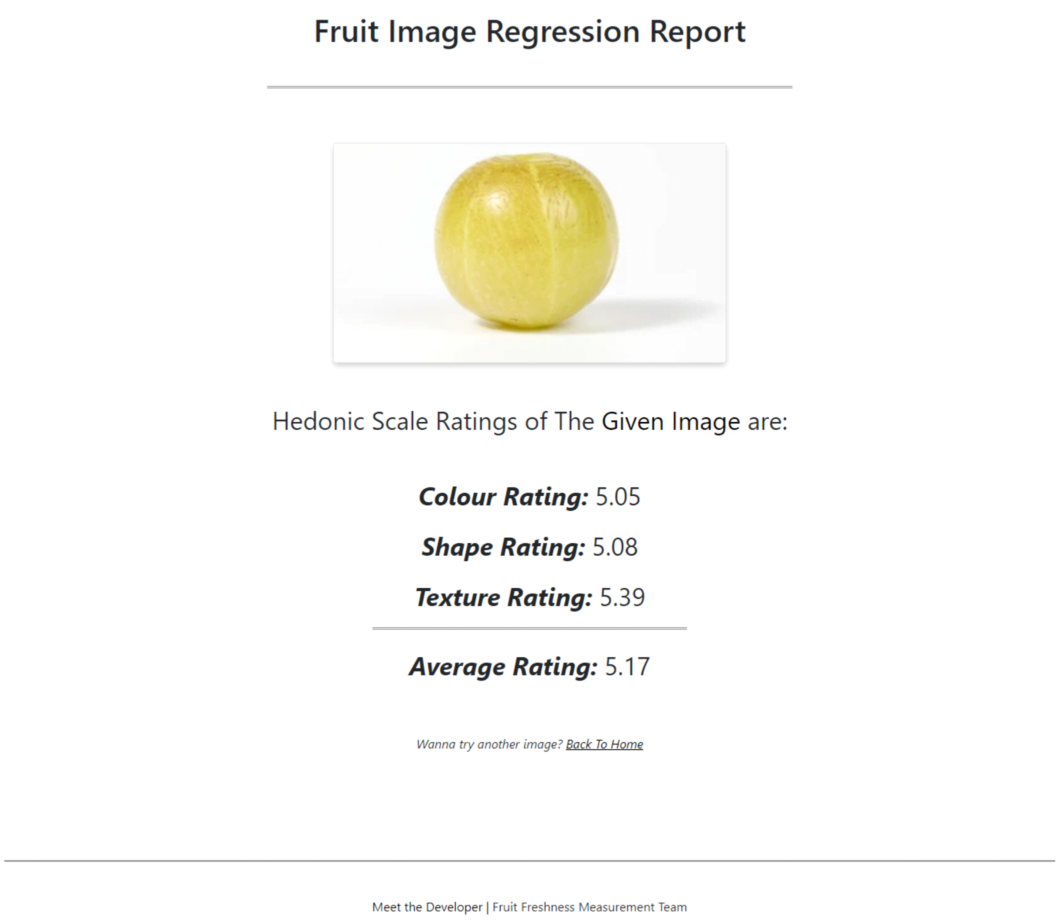

Figure 5.5.4 Regression Task

- **Combined Report: -** The page, after clicking Combined Task, will be as per figure.

Fruit Image Combined Report

1. Classification Report

The Fruit Corresponding To The Given Image Is **FRESH**

2. Regression Report

Hedonic Scale Ratings of The Given Image are:

Colour Rating: 8.5

Shape Rating: 8.51

Texture Rating: 8.5

Average Rating: 8.5

Wanna try another image? Back To Home

Meet the Developer | Fruit Freshness Measurement Team

Figure 5.5.5 Combined Task

Currently, we have not published our Web App in the cloud. It is running on our local server. In future, we will publish it in the cloud so that everyone can use the app.

6 Conclusion and Future Work

6.1 Conclusion

In this project we have created a new amla fruit dataset. We have also created a custom CNN model a DenseNet-121 and a Ridge Regression based model. The performances of the aforementioned neural networks are recorded. the final classification results for both the models show F1 scores of more than **80%** and the FFRM has
RMSE score of **0.04024366303692277.**

6.2 Future Work

To develop a more robust and accurate fruit freshness assessment deep learning model, as a common deep learning practice, a large volume of source data is required. The data should include noises and pictures in different orientations.

In the future a cross platform application can be developed to give users more flexibility to use the app across multiple devices.

7 References

1. Jahanbakhshi, A., Kheiralipour, K., 2019. Carrot sorting based on shape using image processing and artificial intelligent. J. Agric. Mach. 9, 295–307.

2. Izadi, H., Kamgar, S., Raoufat, M.H., 2016. Tomato grading system using machine vision technology and neuro-fuzzy networks (ANFIS). J. Agric. Mach. 6, 49–59

3. Kheiralipour, K., Pormah, A., 2017. Introducing new shape features for classification of cucumber fruit based on image processing technique and artificial neural networks. Food Process Eng. 40, e12558.

4. Wang, S.H., Chen, Y., 2018. Fruit category classification via an eight-layer convolutional neural network with parametric rectified linear unit and dropout technique. Multimed. Tools Appl. 1–17.

5. Jahanbakhshi, Mohammad Momeny, Mahmoudi, Zhang, 2019.Classification of sour lemons based on apparent defects using stochastic pooling mechanism in deep convolutional neural networks

6. M. T. Chowdhury, M. S. Alam, M. A. Hasan and M. I. Khan, Vegetables detection from the glossary shop for the blind, IOSR J. Electr. Electron. Eng. 8 (2013), 43–53.

7. A. Danti, M. Madgi and B. S. Anami, Mean and range color features based identification of common Indian leafy vegetables. Int. J. Sign. Proc. Image Proc. Pattern Recogn. 5 (2012), 151–160.

8. Singla, A., Yuan, L., & Ebrahimi, T. (2016). Food/non-food image classification and food categorization using pre-trained GoogLeNet model.In Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management (pp. 3–11).

9. Sukanya Wichchukit and Michael O’Mahonyc, 2014. The 9-point Hedonic scale and Hedonic ranking in food science: some reappraisals and alternatives.